

Enhance Ahead-Pipelined N-Branch GShare with Tagged Tables

CBP-NG Workshop in Raleigh, NC

Jun Fan

June 28, 2026

Outline

- Background and Motivation
- Predictor Operation
 - Prediction
 - Training
 - Optimization for accuracy, latency and energy
- Results

Outline

- Background and Motivation
- Predictor Operation
 - Prediction
 - Training
 - Optimization for accuracy, latency and energy
- Results

Background

- VFS (Voltage-Frequency-Scaled Speedup) combined by three figures of merit (FoM)
 - IPC: prediction throughput
 - CPI: cycles lost due to misprediction
 - EPI: dynamic energy consumed by the predictor
- Target is to maximize VFS, and needs IPC **↑**, CPI **↓**, EPI **↓**
- CBP-NG simulator
 - Two levels of predictors (P1 and P2)
 - If P2 and P1 disagree, P2 overrides P1, short misprediction happens
 - If P2 mis-predicts, long misprediction happens
 - Example predictors are provided

Motivation

	Advantage	Throughput	Block Size	Ahead-pipeline	Predict Latency	RAM Size	RAM Table Num
tage	lowest MPKI	16 instr/cycle	16 instr	one-block-ahead	P1 = 1, P2 = 2	31 KB	total: 37 gshare: 1 pred, 1 hyst bimodal: 1 pred, 1 hyst tage: 8 tag, 8 pred, 8 hyst, 8 u-bit
gshareN_ahed	highest IPC	7 br/cycle	1024 instr	two-block-ahead	P1 = 1, P2 = 1	128 KB	total: 2 gshare: 1 pred, 1 hyst

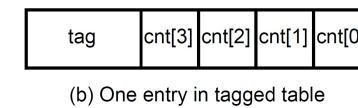
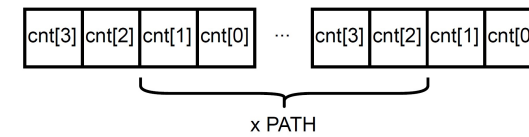
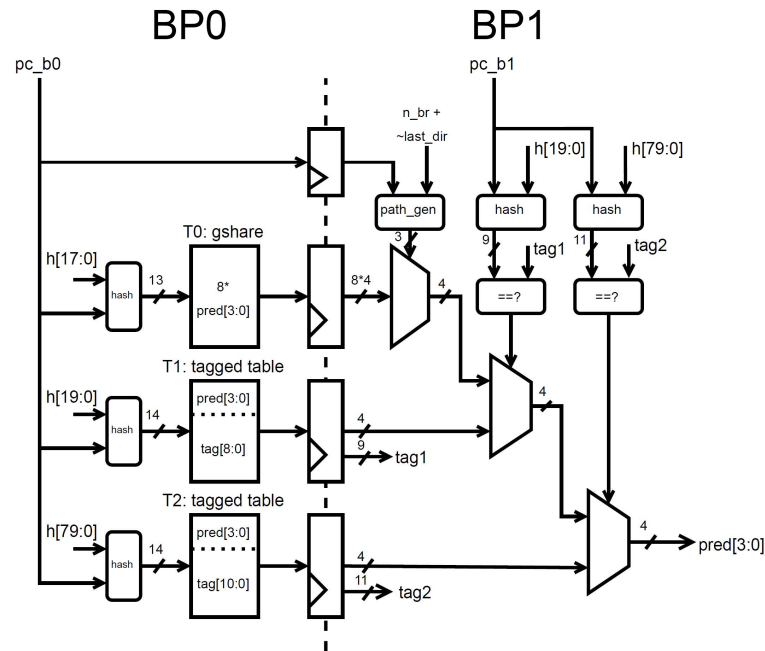
	VFS	IPC	CPI	EPI	MPKI
tage	0.8728	5.769	0.05489	1265.5	5.489
gshareN_ahed	0.9736	9.197	0.06237	243.1	6.930

- VFS ↑ needs IPC ↑, CPI ↓, EPI ↓
 - IPC (gshareN_ahed > tage): larger block size and two-block-ahead (no P2 overrides)
 - CPI (gshareN_ahed > tage): smaller mispredict penalty (no P2) but higher MPKI
 - EPI (gshareN_ahed < tage): less RAM access from larger block size and smaller table number
- Proposed *gshareN_tagtN_ahed*
 - Keep the N-branch style with large block size and two-block-ahead in gshareN_ahed
 - Enhance with tagged tables to lower MPKI so to lower CPI
 - Keep the tagged table number small for low EPI

Outline

- Background and Motivation
- Predictor Operation
 - Prediction
 - Training
 - Optimization for accuracy, latency and energy
- Results

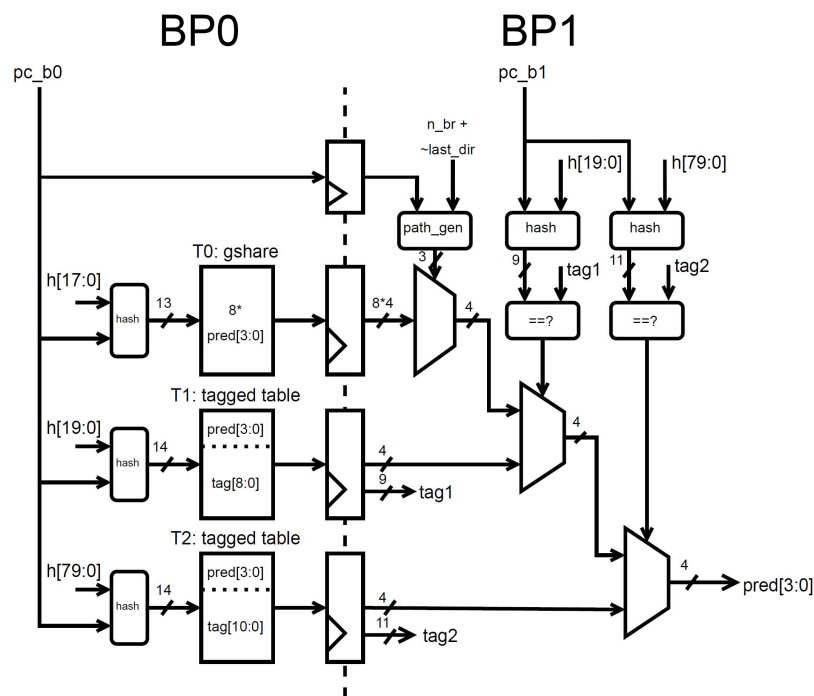
Design Overview



- T0: 8192 entries, $8 * 4 * (3\text{-bit})$ cnt, 1 pred bit, 2 hyst bits
- T1: 16384 entries, 9-bit tag, $4 * (3\text{-bit})$ cnt, 1 pred bit, 2 hyst bits
- T2: 16384 entries, 11-bit tag, $4 * (2\text{-bit})$ cnt, 1 pred bit, 1 hyst bit

	Throughput	Block Size	Ahead-pipeline	Predict Latency
tage	16 instr/cycle	16 instr	one-block-ahead	P1 = 1, P2 = 2
gshareN_ahed	7 br/cycle	1024 instr	two-block-ahead	P1 = 1, P2 = 1
gshareN_tagtN_ahed	4 br/cycle	256 instr	two-block-ahead	P1 = 1, P2 = 1

Predictor Operation: Prediction



- Previous cycle: fetch block pc is pc_b0
 - $\text{hash}(pc_b0, \text{path_history})$ as index to read T0/T1/T2
 - History length for T0/T1/T2 is 18/20/80.
- Current cycle: fetch block pc is pc_b1
 - Select GShare prediction with path
 - Tag match by $\text{hash}(pc_b1, \text{path_history})$
- Final prediction for pc_b1 is from the longest matched table, and it provides predictions for all $N=4$ branches

Predictor Operation: Training

- No misprediction
 - No read to hyst, update hyst of the longest matched table directly
 - T0 to 2'b11, T1 to 2'b10, T2 to 1'b1
 - Prediction+tag bits and hysteresis bits are stored in separate RAMs
 - Write to hyst RAM and read of pred+tag RAM is overlapped, not need extra cycle
- Misprediction
 - The longest matched table
 - Read hyst
 - Prediction bit inverted if hyst is weak (hyst == 0)
 - Decrease hyst by 1
 - New entry allocated if the longest matched table is not T2
 - T1 only allocated if longest matched table is T0
 - T2 always allocated if longest matched table is T0 or T1
 - Need extra one cycle to update pred+tag and hyst RAM

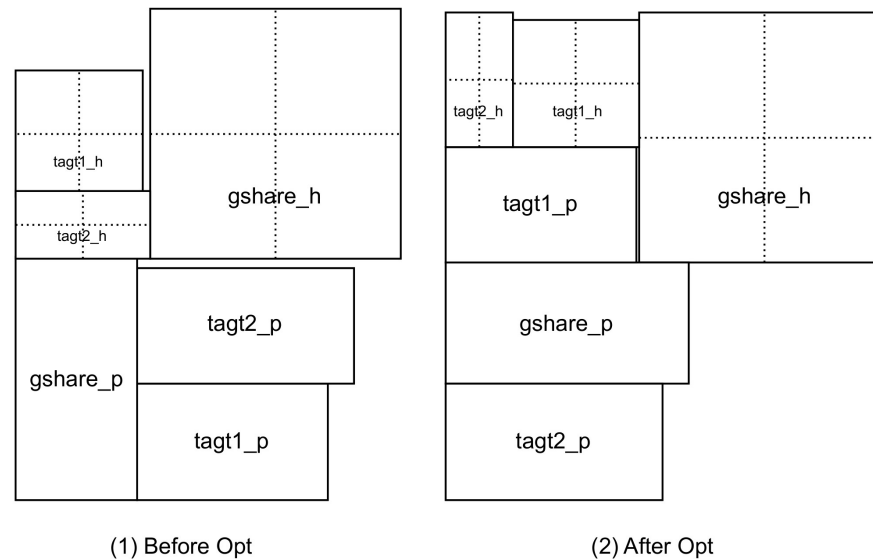
Accuracy Optimizations

- MPKI ↓ leads to CPI ↓ and so VFS ↑
- Recent-PC Training-Direction Bias
 - Track branch PC and direction by two-entry direct-mapped table when longest matched is GShare
 - 3-bit saturating counter, [2] is direction, [1:0] is hysteresis.
 - Longest matched is GShare, mis-predicted PC hits the bias table and hyst is strong (2'b00 or 2'b11), training direction is cnt[2], not actual branch direction.
- Tagged-Table index hashed with path
 - De-alias same consecutive blocks reached by different paths

	CPI	MPKI	RAM Size
gshareN_ahead (CBP-NG example)	0.06237	6.930	128 KB
tage (CBP-NG example)	0.05489	5.489	31 KB
gshareN_ahead (0.5x size)	0.06478	7.198	64 KB
+ two tagged table (T1, T2)	0.04861	5.401	130 KB
+ training-direction bias	0.04838	5.376	130 KB
+ tagt index hash with path	0.04829	5.365	130 KB
+ 2-bit hysteresis	0.04705	5.228	176 KB

Latency Optimization

- Latency ↓ leads to IPC ↑ and so VFS ↑
- RAM floorplan: P1/P2 latency from 1.17 cycle to 0.96 cycle.
 - Before: long routing distance between prediction RAMs
 - After: prediction RAM placed into a more compact layout



Energy Optimizations

- EPI ↓ leads to VFS ↑
- GShare RAM read-enable gated if prediction RAM index is the same for the current and previous block (and training not updated for this index)
 - GShare prediction can be reused from pipeline register, not read RAM again
- T1 index is located near the T1 prediction RAM to reduce wire cost
- Heavy and large gate buffer removed on non-critical path

Energy Optimization	EPI
No EPI opt	320.7
+ GShare RAM read gating	315.5
+ Tagged table index placement	313.7
+ Non-critical path heavy buffer removed	312.6

Outline

- Background and Motivation
- Predictor Operation
 - Prediction
 - Training
 - Optimization for accuracy, latency and energy
- Results

Result

	VFS	IPC	CPI	EPI	MPKI	Critical Path Latency		RAM Size
						P1	P2	
tage (CBP-NG example)	0.8728	5.769	0.05489	1265.5	5.489	0.937	1.863	31 KB
gshareN_ahead (CBP-NG example)	0.9736	9.197	0.06237	243.1	6.930	0.943	0.943	128 KB
tage (CBP2025 TAGE-SC w/o SC)	--	--	--	--	4.430	--	--	142.8 KB
gshareN_tagtN_ahead	0.9921	8.893	0.04705	312.6	5.228	0.96	0.96	176 KB

Thank You